by Rackspace®

for Redis®

Highly Available, Fast, Scalable, Fully Managed

rackspace®

Table of Contents

rackspace

1. Introducing ObjectRocket™ for Redis®

Redis is an open source, in-memory, key-value cache and store database that offers exceptionally high read and write speeds, helping developers deliver application performance that meets the high expectations of today's users.

Unlike earlier key-value data stores such as Memcache, Redis allows for persistence on disk. This enables developers to use it as a standalone database or pair it with on-disk relational databases (such as MySQL) or NoSQL databases (such as MongoDB) when necessary. In either case, Redis is perfect for increasing speed and throughput for workloads with high performance requirements.

Often described as a data structure server, Redis supports several different data structure types including strings, hashes, lists, sets, sorted sets, HyperLogLog and bitstrings. While Redis specializes in quickly retrieving data, doing so requires the appropriate data structure types, all optimized for each specific use case. And although Redis supports proven large-scale deployments like Twitter, Tinder and MapMyFitness, it's still a developing technology.

Given these obstacles, many developers and database administrators discover that fine-tuning Redis proves unexpectedly challenging. Identifying mistakes and making smart adjustments can strain scarce development and IT resources. In addition, finding engineers with the real-world experience required to properly configure and manage Redis can be costly and time-consuming.

ObjectRocket for Redis is a managed database-as-a-service platform built from the ground up to help businesses overcome these challenges. Engineered specifically for Redis, the ObjectRocket platform simplifies application development by offering optimized, preconfigured Redis instances in the cloud, backed by award-winning **Fanatical Support**® from Rackspace.

As a fully managed solution, ObjectRocket offers developers access to Redis experts on staff who manage everything from the infrastructure to the configuration, while also providing access to tools like automated availability.

By combining purpose-built technology with leading Redis expertise, the ObjectRocket platform enables developers to benefit from the speeds, scalability and flexibility of Redis for use cases like full-page caching, session stores, leaderboards and more. It offers:

1. Simplified management
2. Optimal performance
3. Easy, powerful scalability
4. Reliable availability, stability and redundancy
5. Data and application portability
6. Flexible deployment options

Ultimately, ObjectRocket for Redis simplifies application development and enables optimal performance and reliability while freeing technical resources to focus on innovation and creativity rather than database management. This allows enterprises and startups to easily improve the underlying speed of web applications, enabling faster time-to-market and the opportunity to focus on development activities that truly differentiate the business and build a lasting competitive edge.

2. Challenges of Managing, Scaling and Optimizing Redis

Redis holds great promise for developers looking to improve the performance and reliability of their websites and applications. However, maximizing speed and ensuring consistency in the public cloud may present many complex challenges.

TUNING AND ARCHITECTURE MISSTEPS IMPACT PERFORMANCE

Due to its flexibility, Redis can pose significant difficulties for developers who lack expertise in performance tuning and architecture design. Since Redis acts more like a data structure server, it requires developers and engineers to determine how data is best represented — whether in simple key-value strings, lists, hashes, sets or sorted sets. Each data type must be optimized for specific use cases.

For instance, storing hundreds of thousands of jobs in a list and attempting to retrieve jobs in alphabetical order will be frustratingly slow. As another example, the replication timeout in Redis is set to 60 seconds by default, which may be far too long or not long enough when working with slow-performing storage, large data sets or strict network bandwidth limitations.

MANAGING REDIS STRAINS TECHNICAL RESOURCES

Without dedicated engineers with Redis experience, developers may find themselves spending too much time configuring Redis, troubleshooting bottlenecks and rethinking data structure decisions — instead of writing code. In addition to slowing time-to-market, this can reduce application quality and deprive technical contributors of the time they need to pursue new ideas. However, because engineers with Redis experience are in such high demand, finding, hiring and retaining them is time-consuming and expensive.

ACHIEVING HIGH AVAILABILITY IS DIFFICULT

In an increasingly globalized, online world — where even brief periods of downtime can seriously damage both revenue and reputation — high availability has become a requirement for nearly any application. However, manually configuring Redis for high availability and automatic failover is a tedious, highly involved process requiring expertise in the fundamental concepts and techniques of high availability. Redis requires a significant time investment from engineers who have a deep understanding of its complex, distributed Sentinel system — as well as application code, in many cases, to handle failover incidents.

GENERIC PUBLIC CLOUD SERVERS YIELD INCONSISTENT PERFORMANCE

Ensuring fast, consistent database performance is difficult on the public cloud. As with any high I/O database workload, performance on Redis is particularly impacted by noisy neighbor problems (i.e., shared network space). Spikes in multi-tenant utilization not only slow down query speeds, but also make them unpredictable and difficult to code around. As a result, applications that rely on generic public cloud databases experience inconsistent performance. This makes it difficult for businesses to fully realize the performance potential of Redis, which is often the reason they decide to use it in the first place.

SCALABILITY IS RISKY AND EXPENSIVE

Because Redis is an in-memory database, scaling the system to meet growing user demand requires developers to carefully manage memory usage. For example, if a database stores 512 kilobytes of data in Redis for each user and the user base grows by a million users over the course of a month, the database would require the addition of nearly 500 gigabytes of RAM in a short timeframe.

Redis includes built-in, space-saving optimizations, and there are also various tactics available, such as cleaning out unnecessary keys. Many developers lack the time and expertise to take advantage of these measures. As a result, expanding the infrastructure becomes unexpectedly expensive as developers either add multiple Redis servers (and application code), or migrate to larger servers for increased live memory capacity.

3. Benefits of ObjectRocket

ObjectRocket helps businesses resolve these challenges by simplifying database management and ensuring performance on an infrastructure optimized specifically for Redis. By minimizing the hassles, risks and expense of managing Redis in-house, both startups and Fortune 500 enterprises alike can realize faster time-to-market and lower overhead costs. This, in turn, creates a distinct competitive advantage.

"…ObjectRocket for Redis also provides tools and assistance with scaling. Via the ObjectRocket UI, customers can change the size of their instance(s)."

SIMPLIFIED MANAGEMENT

To ease the burden of managing Redis, ObjectRocket offers:

- Configuration and maintenance managed by Redis experts
- **Fanatical Support**® to ensure scalability and optimal performance for every use case
- A suite of custom tools for easy deployment, management and automation

FULLY MANAGED SOLUTION WITH FANATICAL SUPPORT

ObjectRocket provides a fully managed Redis solution backed by industry-leading Fanatical Support, so that Rackspace engineers with Redis experience can enhance your development team's efforts. This allows businesses to free up their internal technical resources so they can focus on innovation and development.

In addition to expert support 24 hours a day, seven days a week, 365 days a year, Fanatical Support provides developers with the assistance they need around architecture design, optimization and configuration. This helps their websites and applications achieve the performance they expect. It also includes assistance with leveraging Redis data types, performance tuning, issue diagnostics and security configurations (access control list, accounts, etc.).

ObjectRocket for Redis also provides tools and assistance with scaling. Via the ObjectRocket UI, customers can change the size of their instance(s). Redis experts can also assist developers with resizing if they need help or have questions.

Fanatical Support includes assistance for disaster recovery planning and business continuity. Support experts assist in locating and accessing a backup file in the event of an instance restoration. Business continuity services also include:

- Advanced administration, monitoring and alerts
- Recommended and planned replication to the disaster recovery site
- Participation in DNS management for failover to the disaster recovery site
- Participation in disaster recovery RTO (recovery time objective) and RPO (recovery point objective) requirements

In addition, ObjectRocket engineers offer guidance to ensure optimal data structure for each use case. For example, there may be cases where using hashes instead of lists can increase performance. Migration services make it easy to get started, and it's all backed by industry-leading SLAs.

Lastly, ObjectRocket experts help developers stay current on Redis. This includes managing updates of Redis versions, patching and direct communication and escalation to Redis on bugs and feature requests.

SUITE OF OPERATIONS TOOLS

The ObjectRocket platform includes a range of back-end functions like automated deployment as well as operations tools that simplify the management of Redis.

Functionality includes:

- **3-click deployment.** Automated deployment enables developers to deploy Redis instances in just three clicks.
- **Easy-to-use interface.** An intuitive user interface enables developers to easily manage their infrastructure and tools.
- **Automated replication.** Spinning up a Redis instance with ObjectRocket includes generation of a second instance which acts as the replica. All master instance data are then replicated to this slave instance.
- **High availability with automatic, nearly transparent failover by default.** Redis Sentinels are used to monitor the instances and, when necessary, initiate a failover process where a slave is promoted to master and applications using the Redis server simply use a standard reconnect process to transparently pick up the change. This requires no Sentinel-aware code in your application.
- **Monitoring, alerting and reporting.** Consistent observation helps ensure ongoing optimal performance.

OPTIMIZED PERFORMANCE

Making applications run fast is one of the major selling points of Redis. Optimizing the infrastructure to realize that benefit is crucial. Every aspect of ObjectRocket, from infrastructure to configuration, has been architected specifically to maximize the fast speeds available with Redis. This resolves the inconsistencies and frustrations that developers typically experience when running databases like Redis on generic public cloud servers. For instance, if a database typically performs at a consistent speed, developers are able to code around it. But if speeds rise and fall unpredictably between two and 1,000 milliseconds on the generic public cloud server, application problems arise.

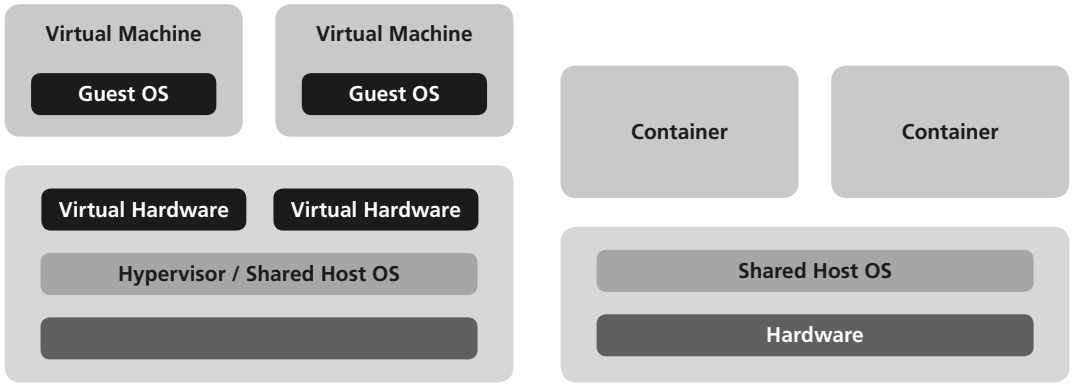
To resolve public cloud inconsistencies, ObjectRocket runs on an infrastructure optimized specifically for Redis performance. As a case in point, the platform uses containers to help eliminate the performance bottlenecks of traditional para-virtualization. By increasing transactions per second and enabling fast, consistent performance, developers can deliver a better experience to their end users.

CONTAINER-BASED VIRTUALIZATION

ObjectRocket's containerized approach isolates CPU, memory and I/O resources for performance improvements. Traditional hardware virtualization — with virtualized memory, processor and hard drives — is less than ideal for high I/O database workloads like Redis. The overhead of running multiple operating systems, sometimes known as the "hypervisor tax," imposes penalties that especially impact database workloads. Additionally, some hypervisors are known to degrade Redis performance in unique ways.

On the other hand, container-based virtualization provides close-to-native performance, eliminating the need for virtualized hardware and multiple operating systems.

Traditional Hardware Virtualization vs. Container-Based Virtualization



ENHANCED LOAD BALANCER

ObjectRocket for Redis includes a load balancer implementation that's targeted specifically for Redis. Based on the open source Redis HAProxy, the built-in load balancer runs on the same performance level servers as ObjectRocket for Redis, resulting in improved performance.

ObjectRocket also ensures that these load balancers are highly available, just like the Redis containers. Load balancing is particularly important for applications in the cloud that require high availability while growing and scaling for increased traffic.

IMPROVED STABILITY AND AVAILABILITY WITH GUARANTEED UPTIME

ObjectRocket for Redis was designed with data stability and accessibility in mind. The architecture of ObjectRocket for Redis provides built-in redundancy of Redis instances with high availability and automatic failover. This in combination with an industry-leading SLA helps ensure that customers have access to their Redis instance and data. Redis has native support for data persistence built in. ObjectRocket for Redis gives customers the capability to manage this data persistence and assurance that their data is safe in the event of any downtime or issues.

SMART PROVISIONING AND FULLY REDUNDANT INFRASTRUCTURE FOR HIGH AVAILABILITY

From the very beginning of the instance being created, ObjectRocket for Redis creates a master and slave pair of the same instance. With two instances, ObjectRocket for Redis ensures high availability through primary and secondary servers. Instances are multiplexed across multiple, physically separate hosts. In addition, ObjectRocket for Redis is built on fully redundant infrastructure — from the network all the way up the stack — further reducing the possibility of downtime.

AUTOMATIC REPLICATION AND FAILOVER

To help ensure stability and availability, ObjectRocket for Redis was designed for data replication and failover. The replication functionality automatically copies the data of the master instance to the slave instance in real time. Object Rocket for Redis ensures that the instance pair (master and slave) stays synchronized. In the event of an issue, Redis Sentinels are used to initiate a seamless failover process where the slave is promoted to the master. Applications using the Redis server are automatically directed to the new address when connecting.

USER-CONTROLLABLE PERSISTENCE

Redis natively offers two methods to ensure permanent access to data for their instances. RDB (Redis database) persistence performs point-in-time snapshots of a dataset at specified intervals, provided enough changes occurred in the specified periods. AOF (append-only files) persistence logs every write operation received by the server. This is disabled by default but can be enabled by the user.

ObjectRocket for Redis supports both of these methods for data persistence and are configurable based upon the customer's needs. In both cases, the instance data is written to disk for availability in the event both master and slave instances fail. When the Redis instances come back up, the master is populated with the data from the latest write to disk. The ObjectRocket support team can help customers with the appropriate persistence settings to ensure that their Redis instances perform optimally.

FREE, AUTOMATED BACKUPS

To help guard against data loss, ObjectRocket offers free, daily backups on persistent Redis instances. This backup capability can be used to:

- Augment existing backup schedules
- Cut backup costs or eliminate the cost completely
- Provide offsite backup for disaster or security breach recovery

Backups are performed every 24 hours. Restoring a database is as simple as contacting ObjectRocket support for loading the backup to the instance.

INDUSTRY-LEADING SLAS

ObjectRocket is backed by one of the industry's most comprehensive service level agreements, ensuring that all instances hosted by Rackspace are available. ObjectRocket for Redis guarantees that primary instances will be available 99.95% of the service year. If Rackspace fails to meet this guarantee, service credits will be awarded as follows:

Monthly Availablitly %	% of monthy bill credits
99.0% – < 99.95%	5%
95.00% – < 99.00%	25%
< 95%	50%

Service credits ensure developers are reimbursed for any possibility of downtime

BUILT-IN SECURITY

To help protect against security threats, the ObjectRocket platform offers a range of security management through customized security tools such as dedicated firewalls and intrusion detection.

Additionally, ObjectRocket maintains a secure-by-default approach by requiring network ACL entries for every instance. ACLSync offers an automated solution for synchronizing an environment's IP addresses with ObjectRocket ACLs. By adding and deleting ACLs on the fly as the environment changes, ACLSync saves developers the trouble of manually managing ObjectRocket network access. Passwords are also required for any client accessing an ObjectRocket for Redis instance.

EASY, POWERFUL SCALABILITY

For any digital presence, scalability is a crucial consideration. With ObjectRocket for Redis, a scaling tool is available to increase the size of the Redis instance(s) based on a customer's timeline and needs. Customers can easily go into the ObjectRocket UI and with a few clicks, make the appropriate change. All of the data migration from the old instance size to the new instance size is completed in an efficient and seamless fashion. Engineering experts are also on hand to help should customers have any questions or need any assistance. With appropriate scalability, developers With appropriate scalability, developers can minimize costs while ensuring their end users achieve the results they expect.

> *"Scalability with ObjectRocket means we actually hit our milestones. They've grown with us as fast as we can, and they've made this process seamless. So beforehand, we were onboarding clusters day in and day out. That's all been offboarded to Rackspace, and it's just invisible success."*
>
> ***– Dane Atkinson***
> *CEO, SumAll*

DATA AND APPLICATION PORTABILITY

The ObjectRocket platform is built on open standards and includes integrations to ensure data and application portability. Because ObjectRocket utilizes an open source, community edition of Redis, data is portable, with no threat of database vendor lock-in.

Application workloads relying on ObjectRocket for Redis can be hosted on the Rackspace Cloud or in AWS regions via AWS Direct Connect while minimizing network lag. With ACLSync, ObjectRocket reduces latency by automatically synchronizing other hosting environments' IP addresses. And with EC2 integration, applications can run on EC2 and connect to the Rackspace Cloud database via AWS Direct Connect.

In addition, ObjectRocket for Redis can be integrated with other databases like traditional relational databases or MongoDB. This allows developers to take advantage of the lightning fast speeds of Redis while integrating with existing applications built on other database technologies.

FLEXIBLE DEPLOYMENT OPTIONS

Characteristic of the ease and flexibility offered by ObjectRocket for Redis, the platform provides multiple deployment options to suit developers' internal IT and security needs. With the latest version of RackConnect®, developers can now access ServiceNet and public IP connections. Typically, ServiceNet IP addresses are inaccessible from the public Internet and are local to each datacenter. With ObjectRocket for Redis, developers can configure their account resources, such as cloud servers and ObjectRocket instances, to utilize a ServiceNet IP address rather than a public IP address. And any traffic that occurs between account resources on the Rackspace Network incurs no additional bandwidth charges.

4. ObjectRocket vs. Unmanaged Hosting

Rackspace is the leader in managed cloud hosting, providing valuable benefits throughout the lifecycle of a variety of cloud-based workloads — especially ones with the performance, scalability and availability requirements of web applications utilizing Redis. The following table outlines the differences between unmanaged Redis hosting and Rackspace's ObjectRocket for Redis:

	Unmanaged	ObjectRocket
Deployment Options	Public cloud with generic commodity servers	Multiple deployment options on hardware optimized for Redis: • Public cloud • Private cloud • Dedicated servers • Hybrid cloud
Deployment Process	Manual deployment: • Manual installation • Manual configuration of storage, networking, security, monitoring and more • Manual testing and optimization with a tradeoff between rapid deployment and future scalability	• 3-click deployment
Performance	• Lower performance on generic commodity servers not optimized for high I/O database workloads • Noisy neighbor problems on public cloud create inconsistency • Must invest in-house technical resources to tune and optimize data structures, architecture, etc.	Consistent database performance on hardware configured and tuned specifically to make Redis run as fast as possible
Scaling	Manual scaling: • Customer handles unique challenges and spikes	• Automated vertical scaling tool to increase instance size in a few clicks • Redis engineers on hand for scaling, unique challenges, and spikes
Monitoring	Developers allocate resources to monitor such things as performance and security	ObjectRocket's Redis experts proactively monitor the network, server, and Redis for such things as connections, lag, status of replicas, size and number of queries.
Availability	Manual high availability (HA). May feature: • Pay for backups • On your own for HA • No SLAs	Automatic high availability and backups: • Automatic HA • Industry-leading SLAs • Data persistence through RDB and AOF files • Free, daily backups
Support	Varies by provider, but may feature: • Limited or nonexistent support • Pay more for additional support and services	• **Fanatical Support**® 24x7x365 • Access to cloud engineers and Redis expertise

rackspace®

5. Common Use Cases

ObjectRocket for Redis helps developers achieve maximum performance with a range of use cases:

- Session stores and caching
- Full-page caching
- Queuing
- Metrics
- Leaderboards and counting
- Publish and subscribe

SESSION STORES AND CACHING

Session caching is one of the most common use cases for Redis. It ensures that applications maintain user data such as session tokens and statistics. Since Redis offers persistence, it's often advantageous over other session stores such as Memcached. In addition to persistence, some other cache backend implementations suffer from serious limitations like inferior scalability and a lack of support for grouping of related cache entries.

The open source content management system Magento uses Redis for session caching. To provide support for stable and scalable cache backend, the latest versions of Magento are turning to Redis.

ObjectRocket for Redis is great for the session caching use case with its persistence support and available Redis expertise. As simple as this use case is, it can get tricky quickly.

The ObjectRocket team has significant experience with session caching and is always available to help customers with any issues that arise. High Availability also plays a very big role in session caching. The last thing that a customer wants is to lose their cache mid-session. ObjectRocket for Redis was architected specifically with high availability in mind. This seamless failover capability assures that a customer's cache won't disappear in the event of an issue.

FULL-PAGE CACHING

Aside from basic session tokens, Redis provides an easy full-page caching platform. It enables items such as CSS layout, auto-completion and top-read articles to load quickly while reducing latency and network traffic. Research shows that users will abandon a site and move to a competitor's site if page load times are delayed by even one second.* Both Magento and WordPress are now available with Redis full-page cache plugins, offering a dramatic increase in page load times.

The way that ObjectRocket for Redis is optimized for performance really shines in this use case. The entire ObjectRocket Redis HW and SW stack was designed specifically for Redis performance and to not slow Redis down. This includes ensuring that the network doesn't get in the way to cause latency and thus slow down the performance. Performance is key in full-page caching and ObjectRocket for Redis is a great tool in this use case.

QUEUING

The Redis in-memory storage engine makes it an appealing platform for job, task and messaging queuing. While many modern web applications try to do everything all at once, queuing can help developers create a more responsive user experience by prioritizing high-visibility jobs and delivering immediate feedback to the user. Related jobs that the user is unlikely to notice can be queued for later execution.

Interacting with Redis as a queue feels native to those who use push/pop operations with lists in programming languages such as Python. There are multiple open source projects that make Redis a back-end utility for all queuing needs. Celery, for example, has a back end that uses Redis as a broker.

While queuing sounds like a secondary use of Redis, it can make all the difference in the world to a web application's performance if used properly. The Redis expertise available with ObjectRocket for Redis — along with how well it performs — makes the platform a good fit for this use case. Our Redis experts can help make recommendations on common approaches to queuing and help to ensure that something that seems secondary doesn't get in the way or become a blocker for the application.

* https://econsultancy.com/blog/10936-site-speed-case-studies-tips-and-tools-for-improving-your-conversion-rate/

METRICS

Measuring database performance and tracking analytics through various metrics is an important consideration for any digital presence. Redis performs well in this capacity due to its speed and flexibility in data structures. It can be used to track metrics such as web traffic, log analysis and in-app usage.

Many customers have some very specific needs from a metrics perspective as it relates to their applications. Redis itself can provide a very easy, flexible way to gather this data in an efficient way without impacting application performance or requiring customers to purchase additional off-the-shelf tools that may or may not fit their needs. If a customer does decide to utilize an off-the-shelf product, the ObjectRocket Redis team is available to help advise and integrate tools with their ObjectRocket for Redis instances.

LEADERBOARDS AND COUNTING

Redis is an appealing technology for websites and applications that involve leaderboards and counting for functions such as ranked commenting system or video game leaderboards. Sets and stored sets perform well in these operations.

The high availability, performance and data persistence of ObjectRocket for Redis make it a good fit for the leaderboard/counting use case as well. Many times, a customer's use of the leaderboard capability of Redis is pivotal to their customer's experience. Much like the caching use case, the last thing that a customer wants is for their leaderboard to disappear or for their leaderboard data to disappear. In the event of an issue, the ObjectRocket for Redis capabilities are key to this use case's success.

PUBLISH AND SUBSCRIBE

In the standard publish and subscribe messaging pattern, message publishers do not program messages to be sent directly to specific subscribers. Instead, published messages are characterized into classes, without consideration of subscribers. Similarly, subscribers express interest in one or more classes and only receive messages that are in that class, without consideration of publishers. The Pub/Sub feature of Redis proves useful for RSS and activity feeds as well as social network connections and even chat systems.

The utilization of Redis in this use case can play a very big role in an application due to its inherent performance. Customers have come to expect an instantaneous experience when it comes to the consumption of information. One of the main goals in the design of ObjectRocket for Redis is to stay out of the way and not slow Redis down. This helps ensure that the flow of information in this use case meets customers' expectations.

6. Conclusion

With its lightning fast in-memory engine, Redis helps deliver the application performance that today's users expect. But for optimal performance, managing and configuring Redis must be done with expert precision.

ObjectRocket for Redis simplifies application development through pre-configuration, performance optimization, reliability and scalability for Redis instances in the cloud. In the end, ObjectRocket for Redis allows developers to focus on innovation and creativity — rather than database architecture — to truly differentiate the business.

To learn more or chat with a specialist: **1-844-208-1147**
or visit **www.objectrocket.com/redis/**

About Rackspace

Rackspace (NYSE: RAX), the #1 managed cloud company, helps businesses tap the power of cloud computing without the challenge and expense of managing complex IT infrastructure and application platforms on their own. Rackspace engineers deliver specialized expertise on top of leading technologies developed by OpenStack®, Microsoft®, VMware® and others, through a results-obsessed service known as **Fanatical Support®**.

GLOBAL OFFICES

Headquarters Rackspace, Inc.
1 Fanatical Place | Windcrest, Texas 78218 | 1-800-961-2888 | Intl: +1 210 312 4700
www.**rackspace**.com

UK Office
Rackspace Ltd.
5 Millington Road
Hyde Park Hayes
Middlesex, UB3 4AZ
Phone: 0800-988-0100
Intl: +44 (0)20 8734 2600
www.rackspace.co.uk

Benelux Office
Rackspace Benelux B.V.
Teleportboulevard 110
1043 EJ Amsterdam
Phone: 00800 8899 00 33
Intl: +31 (0)20 753 32 01
www.rackspace.nl

Hong Kong Office
9/F, Cambridge House, Taikoo Place
979 King's Road,
Quarry Bay, Hong Kong
Sales: +852 3752 6488
Support +852 3752 6464
www.rackspace.com.hk

Australia Office
Rackspace Hosting Australia PTY LTD
Level 1
37 Pitt Street
Sydney, NSW 2000
Australia

Modified: 08272015